1/2

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/545,493	04/07/2000	Rogier Pierre	T2147-906388	9721

7590 01/20/2004

Edward J Kondracki Esq
Kerkam Stowell Kondracki & Clarke P C
1751 PINNACLE DRIVE
SUITE 500
MCLEAN, VA 22102-3833

EXAMINER	
SHAH, NILESH R	
ART UNIT	PAPER NUMBER
2127	10

DATE MAILED: 01/20/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	**Application No.** 09/545,493	**Applicant(s)** PIERRE, ROGIER	
	Examiner Nilesh R Shah	**Art Unit** 2127	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
- Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)□ Responsive to communication(s) filed on 23 October 2003.

2a)☒ This action is **FINAL**. 2b)□ This action is non-final.

3)□ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) 24-49 is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)□ Claim(s) _____ is/are allowed.

6)☒ Claim(s) 24-49 is/are rejected.

7)□ Claim(s) _____ is/are objected to.

8)□ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)□ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on 07 April 1999 is/are: a)☒ accepted or b)□ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)□ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. §§ 119 and 120

12)□ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☒ All b)□ Some * c)□ None of:

1.☒ Certified copies of the priority documents have been received.

2.□ Certified copies of the priority documents have been received in Application No. _____.

3.□ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

13)□ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. § 119(e) (to a provisional application) since a specific reference was included in the first sentence of the specification or in an Application Data Sheet. 37 CFR 1.78.

a) □ The translation of the foreign language provisional application has been received.

14)□ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. §§ 120 and/or 121 since a specific reference was included in the first sentence of the specification or in an Application Data Sheet. 37 CFR 1.78.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) □ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☒ Information Disclosure Statement(s) (PTO-1449) Paper No(s) 2 .

4) □ Interview Summary (PTO-413) Paper No(s). _____ .

5) □ Notice of Informal Patent Application (PTO-152)

6) □ Other: .

DETAILED ACTION

Response to Amendment

This action is in response to amendment filed on October 23, 2003.

Applicant remarks on page 9 state that Carpenter fails to teach the use a number of task queues, which are stored in priority. Carpenter teaches the use of a task queue, which look at the current task priority level to see where the pending queue should be placed. (col. 10 line 30- col. 11 line 17) ('if the input received by IDU 19 is an interrupt request packet issued by an ISU 28, the process proceeds to block 184, which depicts a determination of whether or not the interrupt level specified in the interrupt request packet is (1) greater than the priority level specified in the current task priority register 122 of any processor 10 in the local processing node 8 not currently servicing an interrupt, or (2) high enough to obtain an entry in the pending queue 130 of a processor 10.') ('In addition, as illustrated at block 192, IDU 19 sets a pending flag for the level of the interrupt and sets an active flag for the interrupted processor within the associated current task priority register 122.')

Applicant continues to state that interrupts have nothing to do with tasks. According to Microsoft Computer Dictionary 5th edition, interrupt is defined as a 'single from a device to a computer's processor requesting attention from a process.' It continues to state 'interrupts are the processor's way of communicating with the other elements that make up a computer system'. Clearly interrupts are used to call tasks into action.

Applicant also states Mori does not teach the use of determining the number of processors in each group. Mori specifically teaches the use of determining the number of processes and the statues of each processor (col. 3 line 55 – col. 4 line 24, col. 11 line 32 – col. 13 line 30) ('In the step 835, the processors 11, 13 and 14 have determined that they are not selected to execute the job, and in this case, the timer 106 therein is reset and started for monitoring whether or not the job execution declaration message DCM is normally sent by the selected processor') Mori continues to state that job execution is monitored ('The processor for the job execution is monitored in accordance with whether or not the timer 106 is timed out. In other words, whether or not the job execution declaration was normally effected.'). Thus it is clear by monitoring each processor for job execution the monitor knows the number of processors in teach group.

Finally, applicant states there is no motivation to combine Mori and Carpenter's inventions. Mori teaches the use of monitoring the execution of different jobs in order to make sure each job is running properly. Carpenter teaches task queues are placed in order based on priority. It would be obvious to one skilled in the art to add the teachings of Mori to Carpenter at the time of the invention in order make sure that each processor is able to maximize its performance. By adding the teaching Carpenter and Mori and maximizing performance the entire system is more efficient and able to able more request or tasks.

Thus remarks of applicant are held mute and previous rejection still stands even under new amendment.

Claim Rejections - 35 USC § 102

1. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (e) the invention was described in a patent granted on an application for patent by another filed in the United States before the invention thereof by the applicant for patent, or on an international application by another who has fulfilled the requirements of paragraphs (1), (2), and (4) of section 371(c) of this title before the invention thereof by the applicant for patent.

The changes made to 35 U.S.C. 102(e) by the American Inventors Protection Act of 1999 (AIPA) and the Intellectual Property and High Technology Technical Amendments Act of 2002 do not apply when the reference is a U.S. patent resulting directly or indirectly from an international application filed before November 29, 2000. Therefore, the prior art date of the reference is determined under 35 U.S.C. 102(e) prior to the amendment by the AIPA (pre-AIPA 35 U.S.C. 102(e)).

2. Claims 24-25, 27, 28, 42, 44 and 46 are rejected under 35 U.S.C. 102(e) as being by Carpenter et al (6,148,361) (hereinafter Carpenter).

3. As per claim 24, Carpenter teaches a process for assigning tasks to a processor in a multiprocessor digital data processing system having reemptive operating system, and a given number of processors capable of processing said tasks in parallel, comprising in at least one preliminary phase dividing said processors into groups each group comprising predetermined numbers of processors, dividing said tasks into a predetermined number of elementary task queues and storing a predetermined number of tasks to be processed in a

given priority in each elementary task queue each of said processor groups being associated with an elementary task queue each of the tasks being associated with one of the processors associated with said elementary queue. (col. 3 lines 1-58, col. 8 lines 5-63 and col. 15 lines 7-22) .

4. As per claim 25 Carpenter teaches a process characterized in that said processor groups each comprise an identical number of processors (col. 3 lines 1-58).

5. As per claim 27, Carpenter teaches a process wherein the architecture of said system is of the non-uniform memory access type (NUMA), and the system is constituted by a predetermined number of modules linked to one another, each comprising a given number of processors and storage means, each of said processor modules constituting one of said groups, each module being associated with one of said elementary task queues of an associated processor (col. 3 lines 1-58, col. 8 lines 5-63 and col. 15 lines 7-22).

6. As per claim 28, Carpenter teaches a process further comprising associating each of said processors with a first data structure for identification of the associated processor, said first data structure comprises at least one first set of pointers, associating said first set of pointers with one of said elementary task queues, associating each of said elementary task queues with a second data structure, said second data structure having at least one second set of pointers, associating said second data structure with one of said processor groups, storing all

of the tasks to be processed in said system in a table, each of said second data structures of the elementary queues further comprising a third set of pointers , said third set of pointers each associating elementary task queues with one of said tasks stored in the table or with a series of concatenated tasks, and associating each of said tasks of the table with a third data structure that comprises a fourth set of pointers said fourth set of pointers associating third data structure with one of said elementary queues (col. 3 lines 1-58, col. 8 lines 5-63, col. 14 lines 8- 61 and col. 15 lines 7-22). It is inherent that each task has a pointer associated with where the task is assigned.

7. As per claim 42 Carpenter teaches a architecture for a multiprocessor digital data processing system comprising a given number of processors for implementing a process for assigning tasks to be processed to said processors, said system having a preemptive operating system and a given number of processors capable of processing said task in parallel, said processors being divided into groups, and an elementary queue associated with each of the groups, each of said elementary queues storing a predetermined number of tasks to be processed in a given order of priority, so that each of the tasks of each of said elementary queues is associated with one of the processors of this elementary queue (col. 3 lines 1-58, col. 8 lines 5-63 and col. 15 lines 7-22).

8. As per claim 44, Carpenter teaches an architecture further comprising, when one of said elementary queues associated with one of said processors is empty, means for locating a non-empty, remote elementary queue, and an executable task in said non empty elementary queue, and assigning said executable task to said one of said processor for processing said executable task (col. 3 lines 1-58, col. 8 lines 5-63 and col. 15 lines 7-22).

9. As per claim 46, Carpenter teaches an architecture wherein the operating system of the processing system is of the nonuniform memory access type, and comprises modules linked to one another, each module comprising a given number of processors and storage means, each of said modules constituting one of said groups, each module being associated with one of said elementary queues (col. 3 lines 1-58, col. 8 lines 5-63 and col. 15 lines 7-22).

Claim Rejections - 35 USC § 103

10. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

11. Claim 41 is rejected under 35 U.S.C. 103(a) as being unpatentable in view of Carpenter.

12. As per claim 41, Carpenter teaches the use of an operating system, which can be used with networks (col. 4 line 1-9). Official notice is taken that an operating system called UNIX is well known. It would be obvious to one skilled in the art to use a UNIX as an operating system to provide a source of external communications. UNIX is a widely used as a network operating system.

13. Claims 26 29-32, 36, 37, 39, 40, 43, 45, 47-49 are rejected under 35 U.S.C. 103(a) as being unpatentable over Carpenter as applied to claim 1 above, and further in view of Mori et al (4,462,075) (hereinafter Mori).

14. As per claim 26, Carpenter teaches the use of using multiple processors to complete tasks in parallel but he does not specifically talk about the use of find the processor with the least number of tasks.

Mori teaches a process comprising generating a series of tests and measurements in an additional preliminary phase for determining the number of processors in each group and the number of groups for achieving the best performance of said system (col. 3 line 55 – col. 4 line 24, col. 11 line 32 – col. line 30). It would be obvious to add the teachings of Mori to Carpenter in order make sure that each processor is able to maximize its performance.

15. As per claims 29, Carpenter teaches the use of using multiple processors to complete tasks in parallel but he does not specifically talk about the use of find the processor with the least number of tasks (col. 3 lines 1-58, col. 8 lines 5-63 and col. 15 lines 7-22).

Mori teaches a process comprising distributing said tasks among said elementary queues in at least one additional phase by searching, when a new task to be processed is created, for a queue with the lightest load among all of said elementary queues of said system and assigning said new task to said elementary queue with the lightest load so as to balance the global load of said system among said elementary queues (col. 3 line 55 – col. 4 line 24, col. 11 line 32 – col.13 line 30). It would be obvious to add the teachings of Mori to Carpenter in order make sure that each processor is able to maximize its performance.

16. As per claim 30, Carpenter teaches the use of using multiple processors to complete tasks in parallel but he does not specifically talk about the use of find the processor with the least number of tasks (col. 3 lines 1-58, col. 8 lines 5-63 and col. 15 lines 7-22).

Mori teaches a process further comprising performing said distribution of tasks by determining a composite load parameter associated with each of said elementary queues associating each processor with a memory, calculating said composite load parameter as the sum of the load of a processor or a processor group associated with said elementary queue and the load of the memory associated with said processor or processor group (col. 3 line 55 – col. 4 line 24, col. 11 line 32 – col. 13 line 30). It would be obvious to add the teachings of Mori to Carpenter in order make sure that each processor is able to maximize its performance.

17. As per claim 31, Carpenter teaches the use of using multiple processors to complete tasks in parallel but he does not specifically talk about the use of find the processor with the least number of tasks (col. 3 lines 1-58, col. 8 lines 5-63 and col. 15 lines 7-22).

Mori teaches a process comprising checking in a preliminary step whether said task is linked to one of said elementary queues, and when said test is positive, assigning said linked task to the elementary queue group (col. 3 line 55 – col. 4 line 24, col. 11 line 32 – col. 13 line 30). The test for the lightest load is done before the task is sent to the processor. The processor with the lightest load will receive the task. It would be obvious to add the teachings of Mori to Carpenter in order make sure that each processor is able to maximize its performance

18. As per claim 32, Carpenter teaches the use of using multiple processors to complete tasks in parallel but he does not specifically talk about the use of find the processor with the least number of tasks (col. 3 lines 1-58, col. 8 lines 5-63 and col. 15 lines 7-22).

Mori teaches a process comprising at least one additional phase and searching for a remote elementary queue that is not empty when one of said elementary queues associated with one of said processor groups is empty of executable tasks selecting in said empty elementary queue a task executable by one of said processors of said processor group associated with the empty elementary queue and transmitting said selected task to said one of said processor for processing so as to globally balance the processing of said tasks in said system (col. 3 line 55 – col. 4 line

24, col. 11 line 32 – col. 13 line 30). It would be obvious to add the teachings of Mori to Carpenter in order make sure that each processor is able to maximize its performance.

19. As per claim 36, Carpenter teaches the use of using multiple processors to complete tasks in parallel but he does not specifically talk about the use of find the processor with the least number of tasks (col. 3 lines 1-58, col. 8 lines 5-63 and col. 15 lines 7-22).

Mori teaches a process characterized in that said selected task is associated with a minimal value of a cost parameter, which measures global performance degradation of said system due to the processing of said selected task in said non-empty remote elementary queue by one of said processors of said processor group associated with the empty elementary queue (col. 3 line 55 – col. 4 line 24, col. 11 line 32 – col. 13 line 30). The test for the lightest load is done before the task is sent to the processor. The processor with the lightest load (cost) will receive the task. It would be obvious to add the teachings of Mori to Carpenter in order make sure that each processor is able to maximize its performance.

20. As per claim 37, Carpenter teaches the use of using multiple processors to complete tasks in parallel but he does not specifically talk about the use of find the processor with the least number of tasks (col. 3 lines 1-58, col. 8 lines 5-63 and col. 15 lines 7-22).

Mori teaches a process comprising periodically measuring for a balanced distribution of said tasks in said elementary queues in at least one additional phase and when an unbalanced state of said system is determined, selectively moving tasks from at least one elementary queue with a heavier load to an elementary queue with a lighter load (col. 3 line 55 – col. 4 line 24, col. 11 line 32 – col. 13 line 30). The test for the lightest load is done before the task is sent to the processor. The processor with the lightest load (cost) will receive the task. It would be obvious to add the teachings of Mori to Carpenter in order make sure that each processor is able to maximize its performance and balance the task load.

21. As per claim 39, Carpenter teaches the use of using multiple processors to complete tasks in parallel but he does not specifically talk about the use of find the processor with the least number of tasks (col. 3 lines 1-58, col. 8 lines 5-63 and col. 15 lines 7-22).

Mori teaches a process wherein all or some of said tasks belong to multitask processes, and each multitask process requires a given memory size and workload, further comprising measuring workloads and memory sizes, in the system and selecting the process requiring the greatest workload and the smallest memory size, and moving all the tasks of said selected process to the elementary queue with the lightest load (col. 3 line 55 – col. 4 line 24, col. 11 line 32 – col. 13 line 30). The test for the lightest load is done before the task is sent to the processor. The processor with the lightest load (cost) will receive the task. It would be obvious to add the teachings of Mori to Carpenter in order make sure that each processor is able to maximize its performance and balance the task load.

22. As per claim 40, Carpenter teaches the use of using multiple processors to complete tasks in parallel but he does not specifically talk about the use of find the processor with the least number of tasks (col. 3 lines 1-58, col. 8 lines 5-63 and col. 15 lines 7-22).

Mori teaches a process characterized in that it comprises a preliminary step of checking whether all tasks of said multitask process that must be moved belong to the elementary queue set with the heaviest load and whether any task is linked to any of said groups (col. 3 line 55 – col. 4 line 24, col. 11 line 32 – col. 13 line 30). The test for the lightest load is done before the task is sent to the processor. The processor with the lightest load (cost) will receive the task. It would be obvious to add the teachings of Mori to Carpenter in order make sure that each processor is able to maximize its performance and balance the task load.

23. As per claim 43, Carpenter teaches the use of using multiple processors to complete tasks in parallel but he does not specifically talk about the use of find the processor with the least number of tasks (col. 3 lines 1-58, col. 8 lines 5-63 and col. 15 lines 7-22).

Mori teaches means for determining the load of said elementary queues and for assigning a new task created in said system to the elementary queue with the lightest load (col. 3 line 55 – col. 4 line 24, col. 11 line 32 – col.13 line 30). The test for the lightest load is done before the task is sent to the processor. The processor with the lightest load (cost) will receive the task. It would be obvious to add the teachings of Mori to Carpenter in order make sure that each processor is able to maximize its performance and balance the task load.

24. As per claim 45, Carpenter teaches the use of using multiple processors to complete tasks in parallel but he does not specifically talk about the use of find the processor with the least number of tasks (col. 3 lines 1-58, col. 8 lines 5-63 and col. 15 lines 7-22).

Mori teaches the use of an architecture further comprising means for detecting an imbalance between elementary queues, and for determining when an imbalance is detected the elementary queue with the heaviest load and the elementary queue with the lightest load, and means for moving tasks from the elementary queue with the heaviest load to the elementary queue with the lightest load (col. 3 line 55 – col. 4 line 24, col. 11 line 32 – col.13 line 30). The test for the lightest load is done before the task is sent to the processor. The processor with the lightest load will receive the task. It would be obvious to add the teachings of Mori to Carpenter in order make sure that each processor is able to maximize its performance and balance the task load.

25. As per claim 47, Carpenter teachers an architecture wherein the operating system of the processing system is of the nonuniform memory access type, and comprises modules linked to one another, each module comprising a given number of processors and storage means, each of said modules constituting one of said groups, each module being associated with one of said elementary queues (col. 3 lines 1-58, col. 8 lines 5-63 and col. 15 lines 7-22).

26. As per claim 48, Carpenter teaches an architecture wherein the operating system of the processing system is of the nonuniform memory access type, and comprises modules linked to one another, each module comprising a given number of processors and storage means, each of said modules constituting one of said groups, each module being associated with one of said elementary queues (col. 3 lines 1-58, col. 8 lines 5-63 and col. 15 lines 7-22).

27. As per claim 49, Carpenter teaches an architecture wherein the operating system of the processing system is of the nonuniform memory access type, and comprises modules linked to one another, each module comprising a given number of processors and storage means, each of said modules constituting one of said groups, each module being associated with one of said elementary queues (col. 3 lines 1-58, col. 8 lines 5-63 and col. 15 lines 7-22).

28. Claims 33-35 and 38 are rejected under 35 U.S.C. 103(a) as being unpatentable over Carpenter and Mori as applied to claim 32 above, and further in view of Sutton (5,214,652).

29. As per claim 33, Carpenter teaches the use of using multiple processors to complete tasks in parallel (col. 3 lines 1-58, col. 8 lines 5-63 and col. 15 lines 7-22). Mori teaches the use of the processor with the lightest load (col. 3 line 55 – col. 4 line 24, col. 11 line 32 – col. line 30). Carpenter and Mori do not teach the use of a predetermined threshold.

Sutton teaches process a process comprising discontinuing the step of selectively moving tasks when said imbalance is below a certain threshold (claim 28 and 29). It would be obvious to add the teachings of Sutton to Carpenter and Mori in order to ensure that the processor does not exceed a certain level of process. If the processor were to exceed a certain threshold it would not be efficient thus not completing all tasks.

30. As per claim 34, Carpenter teaches the use of a process further comprising storing the tasks in decreasing order of priority, skipping a predetermined number of tasks before scanning the other tasks of said non-empty elementary queue in order to search for an executable task and have said executable task processed by one of said processors of said processor group associated with the empty elementary queue (col. 3 lines 1-58, col. 8 lines 5-63 and col. 15 lines 7-22).

31. As per claim 35, Carpenter teaches a process characterized in that said number of skipped tasks and the maximum number of scanned tasks among all tasks stored in said non-empty elementary queue are variable over time and are determined by a self-adapting process from the number of tasks that are or are not found during said scans and from the position of these tasks, sequenced in order of priority, in said non-empty elementary queue(col. 3 lines 1-58, col. 8 lines 5-63 and col. 15 lines 7-22).

32. As per claim 38, Carpenter teaches the use of using multiple processors to complete tasks in parallel (col. 3 lines 1-58, col. 8 lines 5-63 and col. 15 lines 7-22). Mori teaches the use of the processor with the lightest load (col. 3 line 55 – col. 4 line 24, col. 11 line 32 – col. line 30). Carpenter and Mori do not teach the use of a predetermined threshold.

Sutton teaches process a process comprising discontinuing the step of selectively moving tasks when said imbalance is below a certain threshold (claim 28 and 29). It would be obvious to add the teachings of Sutton to Carpenter and Mori in order to ensure that the processor does not exceed a certain level of process. If the processor were to exceed a certain threshold it would not be efficient thus not completing all tasks.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Nilesh R Shah whose telephone number is 703-305-8105. The examiner can normally be reached on Monday-Friday 8am-4pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Meng An can be reached on 703-305-9678. The fax phone number for the organization where this application or proceeding is assigned is (703) 872-9306.

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist whose telephone number is 703-305-3900.

MENG-AL T. AN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100

NS

January 7, 2004